

11000 U.S. PTO  
09/888449

# BREVET D'INVENTION

**CERTIFICAT D'UTILITÉ - CERTIFICAT D'ADDITION**

06493/

06/26/01

## COPIE OFFICIELLE

Le Directeur général de l'Institut national de la propriété industrielle certifie que le document ci-annexé est la copie certifiée conforme d'une demande de titre de propriété industrielle déposée à l'Institut.

Fait à Paris, le **10 MAI 2001**

Pour le Directeur général de l'Institut  
national de la propriété industrielle  
Le Chef du Département des brevets

**Martine PLANCHE**

INSTITUT  
NATIONAL DE  
LA PROPRIÉTÉ  
INDUSTRIELLE

SIEGE  
26 bis, rue de Saint Petersburg  
75800 PARIS cedex 08  
Téléphone : 01 53 04 53 04  
Télécopie : 01 42 93 59 30  
<http://www.inpi.fr>

**THIS PAGE BLANK (USP)C,**



26 bis, rue de Saint Pétersbourg  
75800 Paris Cedex 08  
Téléphone : 01 53 04 53 04 Télécopie : 01 42 94 86 54

# BREVET D'INVENTION CERTIFICAT D'UTILITÉ

Code de la propriété intellectuelle - Livre VI



N° 11354\*01

REQUÊTE EN DÉLIVRANCE 1/2

Cet imprimé est à remplir lisiblement à l'encre noire

GB 549 W - 12/02/99

<b>REMISE DES PIÈCES</b> DATE: <b>27 JUIN 2000</b> LIEU: <b>75 INPI PARIS</b> N° D'ENREGISTREMENT NATIONAL ATTRIBUÉ PAR L'INPI: <b>0008243</b> DATE DE DÉPÔT ATTRIBUÉE PAR L'INPI: <b>27 JUIN 2000</b>		<b>1 NOM ET ADRESSE DU DEMANDEUR OU DU MANDATAIRE</b> À QUI LA CORRESPONDANCE DOIT ÊTRE ADRESSÉE COMPAGNIE FINANCIERE ALCATEL Département PI Sylvain CHAFFRAIX 30 avenue Kléber 75116 PARIS	
Vos références pour ce dossier (facultatif): 103102/SYC/MPD/TPM			
Confirmation d'un dépôt par télécopie <input type="checkbox"/> N° attribué par l'INPI à la télécopie			
<b>2 NATURE DE LA DEMANDE</b>		Cochez l'une des 4 cases suivantes	
Demande de brevet		<input checked="" type="checkbox"/>	
Demande de certificat d'utilité		<input type="checkbox"/>	
Demande divisionnaire		<input type="checkbox"/>	
Demande de brevet initiale		N° _____ Date ____/____/____	
ou demande de certificat d'utilité initiale		N° _____ Date ____/____/____	
Transformation d'une demande de brevet européen (Demande de brevet initiale)		<input type="checkbox"/> N° _____ Date ____/____/____	
<b>3 TITRE DE L'INVENTION (200 caractères ou espaces maximum)</b> PROCEDE DE GESTION D'INFORMATIONS EN JAVA			
<b>4 DÉCLARATION DE PRIORITÉ</b> OU REQUÊTE DU BÉNÉFICE DE LA DATE DE DÉPÔT D'UNE DEMANDE ANTÉRIEURE FRANÇAISE		Pays ou organisation _____ N° _____ Date ____/____/____ Pays ou organisation _____ N° _____ Date ____/____/____ Pays ou organisation _____ N° _____ Date ____/____/____ <input type="checkbox"/> S'il y a d'autres priorités, cochez la case et utilisez l'imprimé «Suite»	
<b>5 DEMANDEUR</b>		<input type="checkbox"/> S'il y a d'autres demandeurs, cochez la case et utilisez l'imprimé «Suite»	
Nom ou dénomination sociale		ALCATEL	
Prénoms			
Forme juridique		Société Anonyme	
N° SIREN		5 4 2 0 1 9 0 9 6	
Code APE-NAF			
Adresse		Rue 54, rue La Boétie	
		Code postal et ville 75008 PARIS	
Pays		FRANCE	
Nationalité		Française	
N° de téléphone (facultatif)			
N° de télécopie (facultatif)			
Adresse électronique (facultatif)			

REMISE DES PIÈCES DATE <b>27 JUIN 2000</b> LIEU <b>75 INPI PARIS</b> N° D'ENREGISTREMENT NATIONAL ATTRIBUÉ PAR L'INPI <b>0008243</b>		Réserve à l'INPI	
<b>Vos références pour ce dossier :</b> <i>(facultatif)</i>		103102/SYC/MPD/TPM	
<b>6 MANDATAIRE</b>			
Nom		CHAFFRAIX	
Prénom		Sylvain	
Cabinet ou Société		Compagnie Financière Alcatel	
N° de pouvoir permanent et/ou de lien contractuel		PG 8182	
Adresse	Rue	30 Avenue Kléber	
	Code postal et ville	75116	PARIS
N° de téléphone <i>(facultatif)</i>			
N° de télécopie <i>(facultatif)</i>			
Adresse électronique <i>(facultatif)</i>			
<b>7 INVENTEUR (S)</b>			
Les inventeurs sont les demandeurs		<input type="checkbox"/> Oui <input checked="" type="checkbox"/> Non Dans ce cas fournir une désignation d'inventeur(s) séparée	
<b>8 RAPPORT DE RECHERCHE</b>		Uniquement pour une demande de brevet (y compris division et transformation)	
Établissement immédiat ou établissement différé		<input checked="" type="checkbox"/> <input type="checkbox"/>	
Paiement échelonné de la redevance		Paiement en trois versements, uniquement pour les personnes physiques <input type="checkbox"/> Oui <input checked="" type="checkbox"/> Non	
<b>9 RÉDUCTION DU TAUX DES REDEVANCES</b>		Uniquement pour les personnes physiques <input type="checkbox"/> Requête pour la première fois pour cette invention <i>(joindre un avis de non-imposition)</i> <input type="checkbox"/> Requête antérieurement à ce dépôt <i>(joindre une copie de la décision d'admission pour cette invention ou indiquer sa référence)</i>	
Si vous avez utilisé l'imprimé «Suite», indiquez le nombre de pages jointes			
<b>10 SIGNATURE</b> <del>XXXXXXXXXX</del> <b>XX DU MANDATAIRE</b> (Nom et qualité du signataire)		<b>VISA DE LA PRÉFECTURE OU DE L'INPI</b> M. MARTIN	

DÉPARTEMENT DES BREVETS

26 bis, rue de Saint Pétersbourg  
75800 Paris Cedex 08

Téléphone : 01 53 04 53 04 Télécopie : 01 42 93 59 30

DÉSIGNATION D'INVENTEUR(S) Page N° .1./2..

(Si le demandeur n'est pas l'inventeur ou l'unique inventeur)

Cet imprimé est à remplir lisiblement à l'encre noire

DB 113 W / 263891

Vos références pour ce dossier (facultatif)		103102/SYC/MPD/TPM	
N° D'ENREGISTREMENT NATIONAL		000 8243	
TITRE DE L'INVENTION (200 caractères ou espaces maximum) PROCÉDE DE GESTION D'INFORMATIONS EN JAVA			
LE(S) DEMANDEUR(S) :  Société anonyme <b>ALCATEL</b>			
DESIGNE(NT) EN TANT QU'INVENTEUR(S) : (Indiquez en haut à droite «Page N° 1/1» S'il y a plus de trois inventeurs, utilisez un formulaire identique et numérotez chaque page en indiquant le nombre total de pages).			
Nom		MLYNARCZYK	
Prénoms		Jérôme	
Adresse	Rue	Route de Nozay	
	Code postal et ville	91460   MARCOUSSIS, FRANCE	
Société d'appartenance (facultatif)		ALCATEL CIT	
Nom		CHAFER	
Prénoms		Sylvain	
Adresse	Rue	Route de Nozay	
	Code postal et ville	91460   MARCOUSSIS, FRANCE	
Société d'appartenance (facultatif)		ALCATEL CIT	
Nom		HAUW	
Prénoms		Linda Hélène	
Adresse	Rue	Route de Nozay	
	Code postal et ville	91460   MARCOUSSIS, FRANCE	
Société d'appartenance (facultatif)		ALCATEL CIT	
DATE ET SIGNATURE(S) <del>RIX</del> DU MANDATAIRE (Nom et qualité du signataire)		26 juin 2000 Sylvain CHAFFRAIX	

DÉPARTEMENT DES BREVETS

26 bis, rue de Saint Pétersbourg

75800 Paris Cedex 08

Téléphone : 01 53 04 53 04 Télécopie : 01 42 93 59 30

**DÉSIGNATION D'INVENTEUR(S)** Page N° 2./2..

(Si le demandeur n'est pas l'inventeur ou l'unique inventeur)

Cet imprimé est à remplir lisiblement à l'encre noire

13  
DB 113 W / 2629

<b>Vos références pour ce dossier</b> <i>(facultatif)</i>		103102/SYC/MPD/TPM	
<b>N° D'ENREGISTREMENT NATIONAL</b>		0008243	
<b>TITRE DE L'INVENTION</b> (200 caractères ou espaces maximum) PROCÉDE DE GESTION D'INFORMATIONS EN JAVA			
<b>LE(S) DEMANDEUR(S) :</b>  Société anonyme <b>ALCATEL</b>			
<b>DESIGNE(NT) EN TANT QU'INVENTEUR(S) :</b> (Indiquez en haut à droite «Page N° 1/1» S'il y a plus de trois inventeurs, utilisez un formulaire identique et numérotez chaque page en indiquant le nombre total de pages).			
Nom		MARZOUKI	
Prénoms		Omar	
Adresse	Rue	Route de Nozay	
	Code postal et ville	91460	MARCOUSSIS, FRANCE
Société d'appartenance <i>(facultatif)</i>		ALCATEL CIT	
Nom			
Prénoms			
Adresse	Rue		
	Code postal et ville		
Société d'appartenance <i>(facultatif)</i>			
Nom			
Prénoms			
Adresse	Rue		
	Code postal et ville		
Société d'appartenance <i>(facultatif)</i>			
<b>DATE ET SIGNATURE(S)</b> <del>XX/XX/XX</del> <del>XX/XX/XX</del> <b>DU MANDATAIRE</b> (Nom et qualité du signataire)		26 juin 2000 Sylvain CHAFFRAIX	

## PROCÉDÉ DE GESTION D'INFORMATIONS EN JAVA

L'invention concerne un procédé de gestion d'informations dans un contexte de système réparti et utilisant une méthode d'invocation à distance du langage Java. Le domaine de l'invention est celui de la programmation dans le cadre d'un système réparti et plus spécifiquement de la programmation orientée objet Java utilisant le mécanisme d'invocation de méthode distante ("RMI" en anglais désignant "Remote Method Invocation").

On rappelle qu'en programmation objet, des classes sont définies, chaque classe ayant ses propres caractéristiques. Ces classes sont reliées entre elles par des relations mère-fille, la classe fille héritant des caractéristiques de la classe mère. On distingue parmi ces caractéristiques les "variables" et les "méthodes". Une classe est en quelque sorte un moule à partir duquel on peut créer autant d'objets que nécessaire.

Si l'on considère par exemple la classe voiture V pour laquelle on déclare des "variables" (couleur, nombre de roues, ...) et des "méthodes" (avancer, arrêter, ...), avant de pouvoir utiliser un objet de cette classe, le programmeur devra l'instancier c'est-à-dire le créer par une instruction "new" :

```
v = new V(verte, 4,...)
```

Par cette instruction, le programmeur a créé un objet v aussi appelé instance qui est une voiture verte à 4 roues, .... Il peut alors invoquer les "méthodes" de la classe de cet objet c'est-à-dire les appeler en vue de leur exécution.

Lorsque le programmeur travaille sur un système non réparti, il peut directement instancier des objets comme décrit précédemment et invoquer les "méthodes" de cet objet.

Le langage de programmation orienté objet Java définit également la notion d'interface. Les règles d'héritage pour les interfaces diffèrent de celles des classes, une interface ne définit aucune "variable" non constante et on ne peut instancier une interface comme décrit précédemment. Dans un premier temps, on déclare l'interface et ses "méthodes" mais les "méthodes" sont définies au niveau d'une classe, fille de l'interface. La figure 1 illustre la hiérarchie de cet héritage dans le cas de deux interfaces sœurs A et B dont hérite la classe C « ABImpl » permettant d'implémenter A et B. Les flèches reliant ces classes et interfaces représentent la relation d'héritage.

Les interfaces A et B héritent elles-mêmes d'autres interfaces D et E que l'on ne détaillera pas.

Ainsi si par exemple la "méthode" « afficher » est déclarée pour l'interface A et définie dans la classe C ABImpl, elle pourra être invoquée pour l'objet a de la manière suivante :

```
A a = new ABImpl () ;
a.afficher() ;
```

Pour invoquer la "méthode" « imprimer » de l'interface B, le programmeur pourra passer par l'interface A en utilisant l'instruction nommée « horizontal casting » qui consiste à considérer a comme une instance de la classe B, sœur de la classe A :

```
B b =(B)a ;
b.imprimer() ;
```

La définition sémantique de l'instruction « horizontal casting » est la suivante illustrée par la figure 2 :

étant donné trois classes X, Y, Z et une instance z de la classe Z,  
 Z hérite directement ou indirectement de X et de Y,  
 X n'hérite pas directement ou indirectement de Y,  
 Y n'hérite pas directement ou indirectement de X.

L' « horizontal casting » est l'opération de « casting » sur  $z$  de  $X$  dans  $Y$ .

5 Lorsque le système est réparti, les classes et les interfaces sont définies au niveau d'un serveur central aussi appelé système distant et les instances sont enregistrées dans un système de nommage par un service de nommage établissant une correspondance entre un nom logique et une référence d'objet distant. Dans notre exemple, l'objet  $v$ , après avoir été instancié au niveau du serveur est enregistré dans le système de nommage par  
10 l'instruction :

registry ( $v$ , « autoverte ») ;

Le système de nommage est en général hébergé dans une machine indépendante du serveur. Un système réparti comprend un ou plusieurs serveurs, une machine pour le système de nommage et au moins un système  
15 local situé sur un site client, généralement distant de celui du serveur, le serveur et le système local incluant une ou plusieurs interfaces à travers lesquelles ils communiquent.

Lorsque le programmeur travaille au niveau d'un site client, il ne peut instancier à ce niveau un objet d'une classe définie au niveau du serveur ; il  
20 ne peut utiliser l'instruction new. Il doit récupérer l'objet instancié au niveau du serveur et enregistré dans le système de nommage ; il le récupère auprès du système de nommage à travers sa référence. Dans notre exemple, il récupérera l'objet  $v$  à travers sa référence "autoverte" par l'instruction :

V  $v$  = NamingService.get("autoverte") ;

25 Le programmeur pourra ensuite invoquer les "méthodes" caractéristiques de la classe de cet objet. Ces étapes constituent les principales étapes d'utilisation du système RMI d'invocation de méthode distante par lesquelles un client peut récupérer la référence sur un objet distant à partir d'un système local dans le but de l'utiliser. Le système RMI

comprend notamment une interface RMI et une classe d'implémentation de l'interface RMI.

La figure 3 est une représentation schématique d'un système réparti comprenant un serveur S et un système local Cl. Au niveau du serveur S sont définies en particulier les interfaces RMI, A et B et la classe C ABImpl permettant d'instancier les objets de ces interfaces A et B, comme décrit précédemment. Les interfaces A et B héritent elles-mêmes de l'interface RMI afin que les objets instanciés par le serveur S puissent être récupérés par le système Cl. L'application client « AppClient » consiste en un ensemble d'instructions ; les flèches noires entre « AppClient » et les interfaces A et B signifient que les objets de A et B utilisés dans l'« AppClient » sont récupérés au niveau du serveur S où ils sont définis. Les flèches blanches illustrent des relations d'héritage alors que les flèches noires illustrent des relations d'utilisation.

Mais dans le cas des interfaces d'un système réparti, l'instruction d'« horizontal casting » qui porte alors sur un objet distant c'est-à-dire un objet dont les "méthodes" sont invoquées par l'interface RMI, ne peut être utilisée.

Lorsque le site client dispose de plusieurs interfaces d'utilisation d'objet distant, la référence d'objet distant récupéré à travers l'une d'elles ne peut être considéré comme une référence d'objet valide pour les autres interfaces : pour pouvoir utiliser le même objet distant à partir d'une autre interface, il faut réitérer les étapes RMI décrites précédemment ce qui surcharge le serveur et le service de nommage en nécessitant un enregistrement supplémentaire pour chaque interface supportée.

L'objet de l'invention est de faciliter la tâche du programmeur en rendant la répartition du système transparente. Il pourra ainsi, notamment dans le cas d'interfaces, récupérer une référence sur un objet distant comme il le ferait d'un objet local.

L'invention concerne un procédé de gestion d'informations dans un contexte de système réparti comprenant au moins un système local et un système distant, et utilisant une méthode d'invocation à distance du langage Java ledit langage comportant des instructions et permettant la création  
 5 d'objets à partir de classes d'appartenance ayant entre elles des relations hiérarchiques, principalement caractérisé en ce qu'il consiste à définir dans le système local des classes répliquant la hiérarchie de classes du système distant et comportant des moyens d'accès auxdites classes du système distant en vue de permettre d'utiliser au niveau du système local des  
 10 instructions propres aux classes définies au niveau du système distant.

Selon une caractéristique de l'invention, une instruction est l'instruction « horizontal casting ».

L'invention concerne également un système réparti de gestion d'informations comprenant au moins un système local CI et un système  
 15 distant S comportant plusieurs interfaces A,B et utilisant une méthode d'invocation à distance du langage Java ledit langage comportant des instructions et permettant la création d'objets à partir de classes d'appartenance, caractérisé en ce que le système local CI comprend un « proxy » PA,PB par interface A,B ledit « proxy » PA ou PB étant défini pour  
 20 permettre d'utiliser au niveau du système local CI des instructions propres aux interfaces A,B définies au niveau du système distant S.

D'autres particularités et avantages de l'invention apparaîtront clairement à la lecture de la description faite à titre d'exemple non limitatif et  
 25 en regard des dessins annexés sur lesquels :

la figure 1 représente une hiérarchie d'héritage entre deux interfaces et une classe pour un système non réparti,

la figure 2 illustre la définition sémantique de l'instruction "horizontal casting",

la figure 3 est une représentation schématique des relations au sein d'un système réparti comprenant un serveur S et un système local CI selon l'état de la technique,

la figure 4 est une représentation du même système selon l'invention.

5

L'invention consiste comme représenté figure 4, à interposer entre l'application client « AppClient » et le serveur S, des classes répliquant au niveau du système CI la même hiérarchie d'héritage qu'elles ont entre elles au niveau du serveur S. La classe passerelle PA ou « SmartProxyA » correspond à l'interface A du serveur S et la classe passerelle PB ou « SmartProxyB » à l'interface B. Ces classes PA et PB héritent elles-mêmes d'autres classes. Les classes PA et PB répliquées côté système CI sont locales et indépendantes des interfaces distantes auxquelles elles correspondent. Cela permet de profiter de la flexibilité de la programmation à partir de classes locales.

15

Ces « SmartProxy A et B » PA et PB effectuent par délégation les traitements respectivement de A et B. Elles encapsulent les références sur les objets distants mises à jour dans le système de nommage.

20

Dans le contexte d'un système réparti, le programmeur concevra alors les applications dans un cadre logiciel comprenant une interface de programmation d'application (désignée en anglais "API" ou "Application Programming Interface") et une chaîne d'outils c'est-à-dire que les applications seront développées selon des règles spécifiques en fonction des étapes.

25

On rappelle que lors d'une instanciation, les objets sont initialisés avec des valeurs par défaut ; un constructeur est un mécanisme permettant au programmeur de définir d'autres valeurs d'initialisation.

Un constructeur est ainsi défini dans les classes « SmartProxy A et B » PA et PB en vue de récupérer les références des objets distants via le système de nommage. L'instruction new peut alors être utilisée pour instancier des

objets des classes PA et PB définies localement au niveau du système CI mais représentant les interfaces A et B par délégation.

Les instructions permettant d'invoquer localement dans « ApplClient » la méthode « afficher » de l'interface A définie au niveau du serveur,

```
5      A a = NamingService.get(« référenceA ») ;
      a.afficher () ;
```

deviennent selon l'invention :

```
      PA sPA = new SmartProxyA ;
      sPA.afficher () ;
```

10 Le programmeur pourra ensuite utiliser l' « horizontal casting » entre les classes PA et PB définies localement pour invoquer une méthode de l'interface B définie au niveau du serveur, par exemple la méthode « imprimer » :

```
      PB sPB = (PB)sPA ;
15      SPB.imprimer () ;
```

## REVENDEICATIONS

1. Procédé de gestion d'informations dans un contexte de système réparti comprenant au moins un système local et un système distant, et utilisant une méthode d'invocation à distance du langage Java ledit langage comportant des instructions et permettant la création d'objets à partir de classes d'appartenance ayant entre elles des relations hiérarchiques, caractérisé en ce qu'il consiste à définir dans le système local des classes répliquant la hiérarchie de classes du système distant et comportant des moyens d'accès aux dites classes du système distant en vue de permettre d'utiliser au niveau du système local des instructions propres aux classes définies au niveau du système distant.

2. Procédé selon la revendication précédente caractérisé en ce qu'une instruction est l'instruction « horizontal casting ».

3. Système réparti de gestion d'informations comprenant au moins un système local (Cl) et un système distant (S) comportant plusieurs interfaces (A,B) et utilisant une méthode d'invocation à distance du langage Java ledit langage comportant des instructions et permettant la création d'objets à partir de classes d'appartenance, caractérisé en ce que le système local (Cl) comprend un « proxy » (PA,PB) par interface (A,B) ledit « proxy » (PA ou PB) étant défini pour permettre d'utiliser au niveau du système local (Cl) des instructions propres aux interfaces (A,B) définies au niveau du système distant (S).

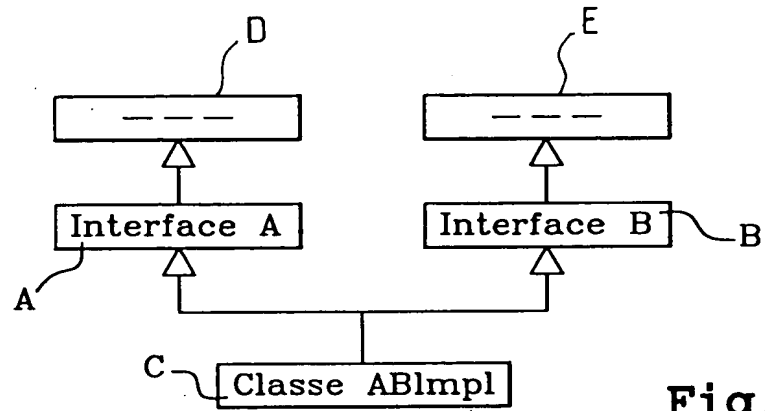


Fig. 1

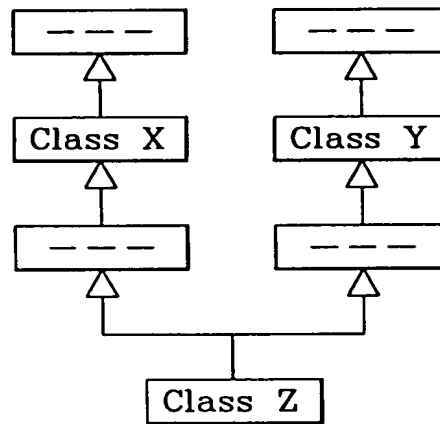
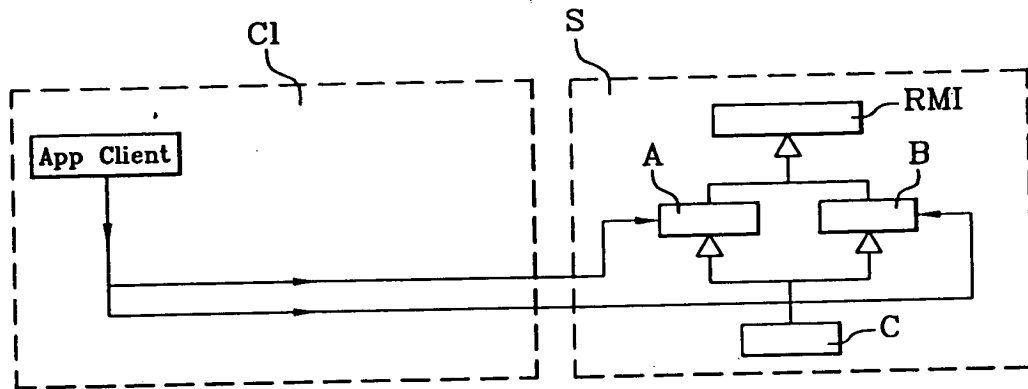
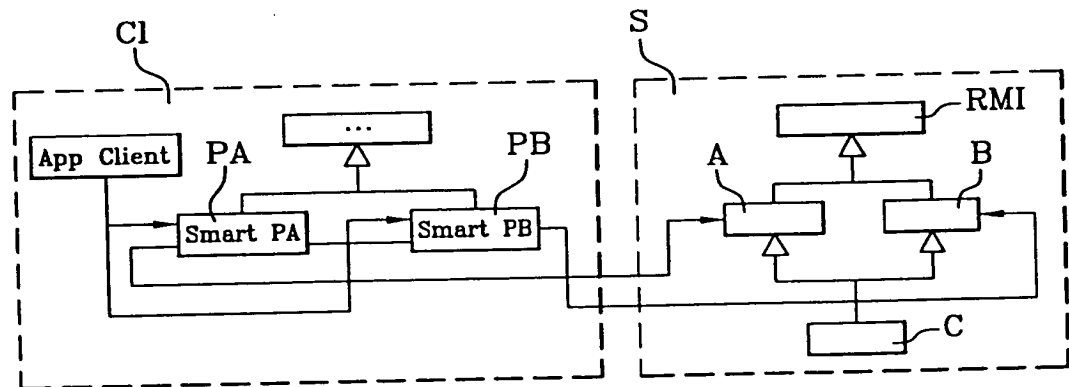


Fig. 2

Fig. 3Fig. 4